Installing packages

In [5]:
```
#much of this tutorial is derived from the scanpy tutorial here:
#https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html

#getting packages that you need for this analysis using the linux package manager pip
!pip install scanpy  #the package we will be using for the bulk of our ana
lysis
!pip3 install leidenalg #the algorithm we will use for clustering
```

```
Collecting scanpy
  Downloading scanpy-1.8.1-py3-none-any.whl (2.0 MB)
     |████████████████████████████████| 2.0 MB 5.2 MB/s eta 0:00:01
Collecting tqdm
  Downloading tqdm-4.62.3-py2.py3-none-any.whl (76 kB)
     |████████████████████████████████| 76 kB 6.6 MB/s  eta 0:00:01
Collecting natsort
  Downloading natsort-7.1.1-py3-none-any.whl (35 kB)
Collecting anndata>=0.7.4
  Downloading anndata-0.7.6-py3-none-any.whl (127 kB)
     |████████████████████████████████| 127 kB 70.7 MB/s eta 0:00:01
Collecting sinfo
  Downloading sinfo-0.3.4.tar.gz (24 kB)
Collecting importlib_metadata>=0.7
  Downloading importlib_metadata-4.8.1-py3-none-any.whl (17 kB)
Collecting scikit-learn>=0.22
  Downloading scikit_learn-0.24.2-cp37-cp37m-manylinux2010_x86_64.whl (22.3 MB)
     |████████████████████████████████| 22.3 MB 50.0 MB/s eta 0:00:01
Collecting umap-learn>=0.3.10
  Downloading umap-learn-0.5.1.tar.gz (80 kB)
     |████████████████████████████████| 80 kB 12.8 MB/s eta 0:00:01
Collecting packaging
  Downloading packaging-21.0-py3-none-any.whl (40 kB)
     |████████████████████████████████| 40 kB 8.1 MB/s  eta 0:00:01
Collecting scipy>=1.4
  Downloading scipy-1.7.1-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (28.5 MB)
     |████████████████████████████████| 28.5 MB 68.6 MB/s eta 0:00:01
Collecting pandas>=0.21
  Downloading pandas-1.3.3-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.3 MB)
     |████████████████████████████████| 11.3 MB 63.5 MB/s eta 0:00:01
Collecting h5py>=2.10.0
  Downloading h5py-3.4.0-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (4.1 MB)
     |████████████████████████████████| 4.1 MB 64.3 MB/s eta 0:00:01
Collecting seaborn
  Downloading seaborn-0.11.2-py3-none-any.whl (292 kB)
     |████████████████████████████████| 292 kB 74.3 MB/s eta 0:00:01
Collecting statsmodels>=0.10.0rc2
  Downloading statsmodels-0.13.0rc0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (9.8
MB)
     |████████████████████████████████| 9.8 MB 27.8 MB/s eta 0:00:01
Collecting patsy
  Downloading patsy-0.5.1-py2.py3-none-any.whl (231 kB)
```

```
     |████████████████████████████████| 231 kB 75.7 MB/s eta 0:00:01
Collecting networkx>=2.3
  Downloading networkx-2.6.3-py3-none-any.whl (1.9 MB)
     |████████████████████████████████| 1.9 MB 59.8 MB/s eta 0:00:01
Collecting numpy>=1.17.0
  Downloading numpy-1.21.2-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.7 MB)
     |████████████████████████████████| 15.7 MB 64.0 MB/s eta 0:00:01
Collecting matplotlib>=3.1.2
  Downloading matplotlib-3.4.3-cp37-cp37m-manylinux1_x86_64.whl (10.3 MB)
     |████████████████████████████████| 10.3 MB 63.1 MB/s eta 0:00:01
Collecting numba>=0.41.0
  Downloading numba-0.54.0-2-cp37-cp37m-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (3.4 MB)
     |████████████████████████████████| 3.4 MB 67.1 MB/s eta 0:00:01
Collecting tables
  Downloading tables-3.6.1-cp37-cp37m-manylinux1_x86_64.whl (4.3 MB)
     |████████████████████████████████| 4.3 MB 64.8 MB/s eta 0:00:01
Collecting joblib
  Downloading joblib-1.0.1-py3-none-any.whl (303 kB)
     |████████████████████████████████| 303 kB 74.5 MB/s eta 0:00:01
Collecting xlrd<2.0
  Downloading xlrd-1.2.0-py2.py3-none-any.whl (103 kB)
     |████████████████████████████████| 103 kB 74.6 MB/s eta 0:00:01
Collecting cached-property
  Downloading cached_property-1.5.2-py2.py3-none-any.whl (7.6 kB)
Collecting zipp>=0.5
  Downloading zipp-3.5.0-py3-none-any.whl (5.7 kB)
Collecting typing-extensions>=3.6.4
  Downloading typing_extensions-3.10.0.2-py3-none-any.whl (26 kB)
Collecting python-dateutil>=2.7
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     |████████████████████████████████| 247 kB 79.3 MB/s eta 0:00:01
Collecting cycler>=0.10
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting pillow>=6.2.0
  Downloading Pillow-8.3.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.0 MB)
     |████████████████████████████████| 3.0 MB 63.0 MB/s eta 0:00:01
Collecting pyparsing>=2.2.1
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
     |████████████████████████████████| 67 kB 8.4 MB/s  eta 0:00:01
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.3.2-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (1.1 MB)
     |████████████████████████████████| 1.1 MB 68.7 MB/s eta 0:00:01
Collecting six
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting llvmlite<0.38,>=0.37.0rc1
  Downloading llvmlite-0.37.0-cp37-cp37m-manylinux2014_x86_64.whl (26.3 MB)
     |████████████████████████████████| 26.3 MB 64.0 MB/s eta 0:00:01
Collecting setuptools
  Downloading setuptools-58.0.4-py3-none-any.whl (816 kB)
     |████████████████████████████████| 816 kB 61.6 MB/s eta 0:00:01
Collecting numpy>=1.17.0
  Downloading numpy-1.20.3-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.3 MB)
     |████████████████████████████████| 15.3 MB 66.8 MB/s eta 0:00:01
Collecting pytz>=2017.3
  Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
     |████████████████████████████████| 510 kB 70.9 MB/s eta 0:00:01
Collecting threadpoolctl>=2.0.0
```

```
  Downloading threadpoolctl-2.2.0-py3-none-any.whl (12 kB)
Collecting pynndescent>=0.5
  Downloading pynndescent-0.5.4.tar.gz (1.1 MB)
     |████████████████████████████████| 1.1 MB 64.4 MB/s eta 0:00:01
Collecting stdlib_list
  Downloading stdlib_list-0.8.0-py3-none-any.whl (63 kB)
     |████████████████████████████████| 63 kB 3.2 MB/s  eta 0:00:01
Collecting numexpr>=2.6.2
  Downloading numexpr-2.7.3-cp37-cp37m-manylinux2010_x86_64.whl (471 kB)
     |████████████████████████████████| 471 kB 73.2 MB/s eta 0:00:01
Building wheels for collected packages: umap-learn, pynndescent, sinfo
  Building wheel for umap-learn (setup.py) ... done
  Created wheel for umap-learn: filename=umap_learn-0.5.1-py3-none-any.whl size=76566 sha256=7673e9c8b
55e7a7a3bd626c57be341a79aacbaf35944058ecd6b01ae19955d4a
  Stored in directory: /home/jupyter/.cache/pip/wheels/01/e7/bb/347dc0e510803d7116a13d592b10cc68262da56
a8eec4dd72f
  Building wheel for pynndescent (setup.py) ... done
  Created wheel for pynndescent: filename=pynndescent-0.5.4-py3-none-any.whl size=52372 sha256=94909d90
b75379ca634c4d9930d188a8b7a977a0e648d9807e374b2228a3220d
  Stored in directory: /home/jupyter/.cache/pip/wheels/d0/5b/62/3401692ddad12324249c774c4b15ccb04694602
1e2b581c043
  Building wheel for sinfo (setup.py) ... done
  Created wheel for sinfo: filename=sinfo-0.3.4-py3-none-any.whl size=7899 sha256=cf86aa1e3999f749bcbb7e
ed9177ca2071f6f9379573403af03c0fc211507a59
  Stored in directory: /home/jupyter/.cache/pip/wheels/68/ca/56/344d532fe53e855ccd6549795d370588ab81239
07eecf4cf30
Successfully built umap-learn pynndescent sinfo
Installing collected packages: numpy, threadpoolctl, six, setuptools, scipy, llvmlite, joblib, zipp, typing-extensio
ns, scikit-learn, pytz, python-dateutil, pyparsing, pillow, numba, kiwisolver, cycler, cached-property, xlrd, stdlib
-list, pynndescent, patsy, pandas, packaging, numexpr, natsort, matplotlib, importlib-metadata, h5py, umap-lear
n, tqdm, tables, statsmodels, sinfo, seaborn, networkx, anndata, scanpy
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This
behaviour is the source of the following dependency conflicts.
tensorflow 2.4.2 requires h5py~=2.10.0, but you have h5py 3.4.0 which is incompatible.
tensorflow 2.4.2 requires numpy~=1.19.2, but you have numpy 1.20.3 which is incompatible.
tensorflow 2.4.2 requires six~=1.15.0, but you have six 1.16.0 which is incompatible.
tensorflow 2.4.2 requires typing-extensions~=3.7.4, but you have typing-extensions 3.10.0.2 which is incompati
ble.
jupyterlab-git 0.11.0 requires nbdime<2.0.0,>=1.1.0, but you have nbdime 3.1.0 which is incompatible.
Successfully installed anndata-0.7.6 cached-property-1.5.2 cycler-0.10.0 h5py-3.4.0 importlib-metadata-4.8.1 jo
blib-1.0.1 kiwisolver-1.3.2 llvmlite-0.37.0 matplotlib-3.4.3 natsort-7.1.1 networkx-2.6.3 numba-0.54.0 numexpr
-2.7.3 numpy-1.20.3 packaging-21.0 pandas-1.3.3 patsy-0.5.1 pillow-8.3.2 pynndescent-0.5.4 pyparsing-2.4.7 p
ython-dateutil-2.8.2 pytz-2021.1 scanpy-1.8.1 scikit-learn-0.24.2 scipy-1.7.1 seaborn-0.11.2 setuptools-58.0.4 si
nfo-0.3.4 six-1.16.0 statsmodels-0.13.0rc0 stdlib-list-0.8.0 tables-3.6.1 threadpoolctl-2.2.0 tqdm-4.62.3 typing-e
xtensions-3.10.0.2 umap-learn-0.5.1 xlrd-1.2.0 zipp-3.5.0
WARNING: You are using pip version 21.1.3; however, version 21.2.4 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pip' command.
Collecting leidenalg
  Downloading leidenalg-0.8.7-cp37-cp37m-manylinux2010_x86_64.whl (1.4 MB)
     |████████████████████████████████| 1.4 MB 5.1 MB/s eta 0:00:01
Collecting python-igraph>=0.9.0
  Downloading python_igraph-0.9.6-cp37-cp37m-manylinux2010_x86_64.whl (3.2 MB)
     |████████████████████████████████| 3.2 MB 62.7 MB/s eta 0:00:01
Collecting texttable>=1.6.2
  Downloading texttable-1.6.4-py2.py3-none-any.whl (10 kB)
Installing collected packages: texttable, python-igraph, leidenalg
Successfully installed leidenalg-0.8.7 python-igraph-0.9.6 texttable-1.6.4
```

```
WARNING: Target directory /home/jupyter/notebooks/packages/__pycache__ already exists. Specify --upgrade
 to force replacement.
WARNING: Target directory /home/jupyter/notebooks/packages/bin already exists. Specify --upgrade to force r
eplacement.
WARNING: You are using pip version 21.1.3; however, version 21.2.4 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pip' command.
```

Getting the single cell data from a google bucket

In [2]:
```
#going and getting the h5ad file, processed through cumulus by Nathan Tucker's comp bio person
#cumulus workspace available here: https://app.terra.bio/#workspaces/kco-tech/Cumulus

#this is from a human aorta data set, we got this dataset from the human cell atlas:
#https://data.humancellatlas.org/explore/projects/07073c12-8006-4710-a00b-23abdb814904

!mkdir data #making a storage directory to put the data!

#here we actually go and get the data.  gsutil is a google cloud utility to copy data from
#"buckets" into our local disk space.  Then we will be able to ingest and manipulate the data
!gsutil cp gs://fc-9c9cf8d0-aafd-47c2-ab1b-4e0f20ab200b/aortic_adata_full_
li_lemaire.h5ad data/
```

```
Copying gs://fc-9c9cf8d0-aafd-47c2-ab1b-4e0f20ab200b/aortic_adata_full_li_lemaire.h5ad...
/ [1 files][  4.7 GiB/  4.7 GiB]   59.2 MiB/s
Operation completed over 1 objects/4.7 GiB.
```

In [42]:
```
#listing the contents of our data directory
!ls data/
```

```
aortic_adata_full_li_lemaire.h5ad
```

Setting up the python environment

In [43]:
```
#importing necessary python packages
import numpy as np
import pandas as pd
import scanpy as sc

#setting parameters for scanpy
sc.settings.verbosity = 3                   # verbosity: errors (0), warnings (1), info (2), hints (3)
sc.logging.print_versions()  #version checking.....
sc.settings.set_figure_params(dpi=80)
```

```
WARNING: If you miss a compact list, please try `print_header`!
The `sinfo` package has changed name and is now called `session_info` to become more discoverable and self-e
xplanatory. The `sinfo` PyPI package will be kept around to avoid breaking old installs and you can downgrade
 to 0.3.2 if you want to use it without seeing this message. For the latest features and bug fixes, please install `ses
sion_info` instead. The usage and defaults also changed slightly, so please review the latest README at https://
gitlab.com/joelostblom/session_info.
-----
anndata     0.7.6
```

```
scanpy      1.8.1
sinfo       0.3.4
-----
PIL                         8.2.0
attr                        21.2.0
backcall                    0.2.0
beta_ufunc                  NA
binom_ufunc                 NA
bottleneck                  1.3.2
brotli                      NA
cached_property             1.5.2
cachetools                  4.2.2
caip_notebooks_serverextension      NA
certifi                     2021.05.30
cffi                        1.14.5
chardet                     4.0.0
cloudpickle                 1.6.0
colorama                    0.4.4
cryptography                3.4.7
cycler                      0.10.0
cython_runtime              NA
dateutil                    2.8.1
decorator                   5.0.9
google                      NA
greenlet                    1.1.0
grpc                        1.32.0
h5py                        3.4.0
idna                        2.10
igraph                      0.9.6
ipykernel                   5.5.5
ipython_genutils            0.2.0
ipywidgets                  7.6.3
jedi                        0.18.0
jinja2                      3.0.1
joblib                      1.0.1
jsonschema                  3.2.0
jwt                         2.1.0
kiwisolver                  1.3.1
leidenalg                   0.8.7
llvmlite                    0.37.0
markupsafe                  2.0.1
matplotlib                  3.4.2
matplotlib_inline           NA
mpl_toolkits                NA
natsort                     7.1.1
nbformat                    5.1.3
nbinom_ufunc                NA
numba                       0.54.0
numexpr                     2.7.3
numpy                       1.19.5
packaging                   20.9
pandas                      1.2.4
parso                       0.8.2
pexpect                     4.8.0
pickleshare                 0.7.5
pkg_resources               NA
prettytable                 2.1.0
```

```
prometheus_client               NA
prompt_toolkit                3.0.19
proto                          NA
psutil                        5.8.0
ptyprocess                      0.7.0
pvectorc                       NA
pyarrow                        4.0.1
pygments                       2.9.0
pyparsing                      2.4.7
pyrsistent                     NA
pytz                          2021.1
requests                       2.25.1
scipy                         1.7.1
seaborn                        0.11.2
send2trash                     NA
simplejson                     3.17.2
six                           1.15.0
sklearn                        0.24.2
socks                         1.7.1
sql                           NA
sqlalchemy                      1.4.18
sqlparse                       0.4.1
statsmodels                     0.13.0rc0
storemagic                     NA
tables                        3.6.1
terminado                      0.10.1
texttable                     1.6.4
tornado                       6.1
tqdm                          4.61.1
traitlets                     5.0.5
typing_extensions               NA
uritemplate                    3.0.1
urllib3                       1.26.5
wcwidth                        0.2.5
yaml                          5.4.1
zipp                          NA
zmq                           22.1.0
-----
IPython             7.24.1
jupyter_client      6.1.12
jupyter_core        4.7.1
jupyterlab          1.2.16
notebook            6.4.0
-----
Python 3.7.10 | packaged by conda-forge | (default, Feb 19 2021, 16:07:37) [GCC 9.3.0]
Linux-5.4.104+-x86_64-with-debian-buster-sid
8 logical CPU cores, x86_64
-----
Session information updated at 2021-09-21 16:35
```

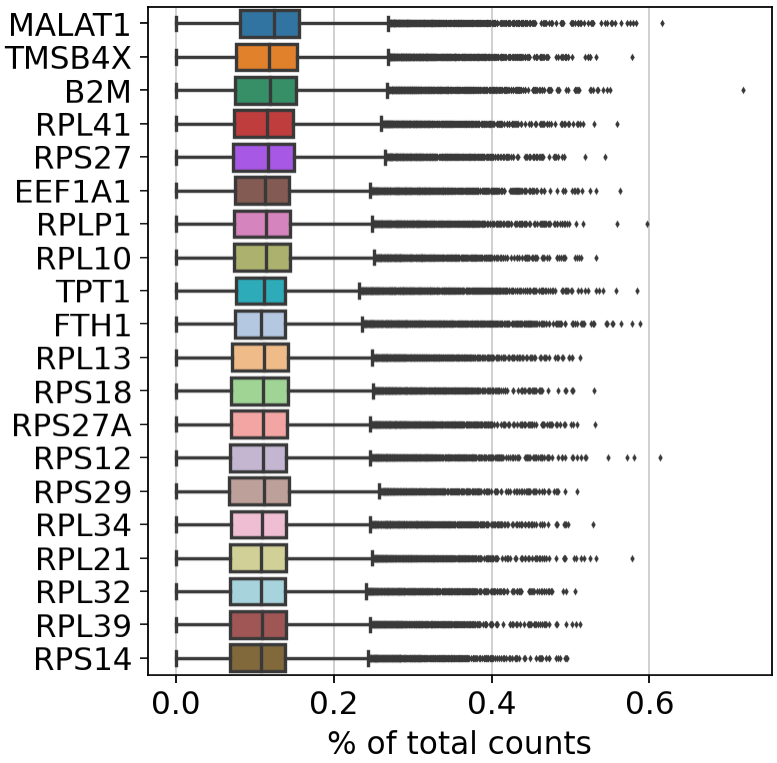
Reading in & processing the single cell aorta data h5ad file

In [44]:
```
#reading in the data using the scanpy function sc.read
adata = sc.read("data/aortic_adata_full_li_lemaire.h5ad") #this file is huge, will take a minute to read in (4.7Gb)
adata #this will show properties of the adata object, a very useful first step
```

Out[44]:
```
AnnData object with n_obs × n_vars = 27225 × 23318
    obs: 'Sample', 'Treatment', 'batch', 'n_genes', 'n_genes_by_counts', 'total_counts', 'total_counts_mt', 'pct_counts_mt', 'leiden'
    var: 'mt', 'gene_ids', 'n_cells_by_counts', 'mean_counts', 'pct_dropout_by_counts', 'total_counts', 'highly_variable', 'means', 'dispersions', 'dispersions_norm'
    uns: 'Sample_colors', 'Treatment_colors', 'hvg', 'leiden', 'leiden_colors', 'neighbors', 'pca', 'umap'
    obsm: 'X_pca', 'X_umap'
```

In [45]:
```
#Show those genes that yield the highest fraction of counts in each single cell, across all cells.
sc.pl.highest_expr_genes(adata, n_top=20, )
```

```
normalizing counts per cell
    finished (0:00:01)
```



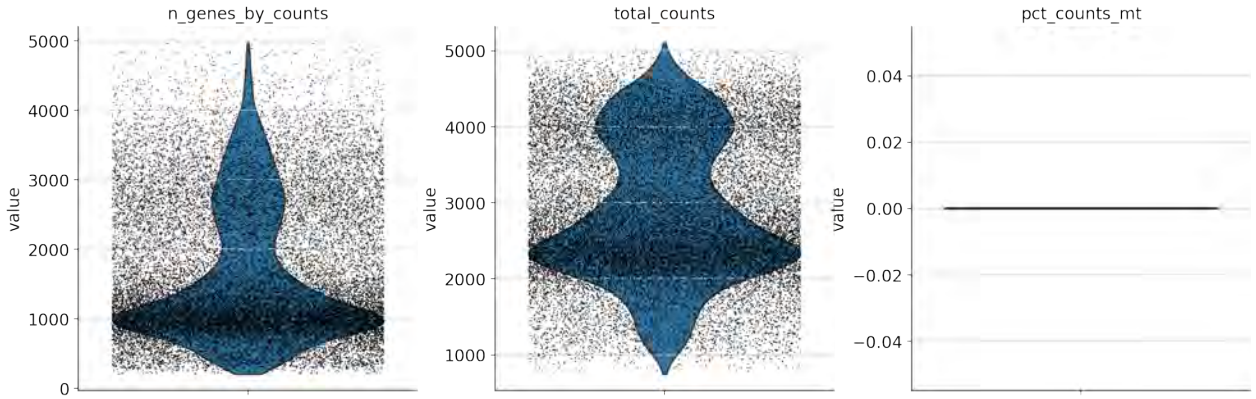
Some filtering of the data

In [46]:
```
#basic filtering:
#cells need to have at least 200 genes expressed
#genes need to be expressed in at least 3 cells

sc.pp.filter_cells(adata, min_genes=200)
sc.pp.filter_genes(adata, min_cells=3)
```

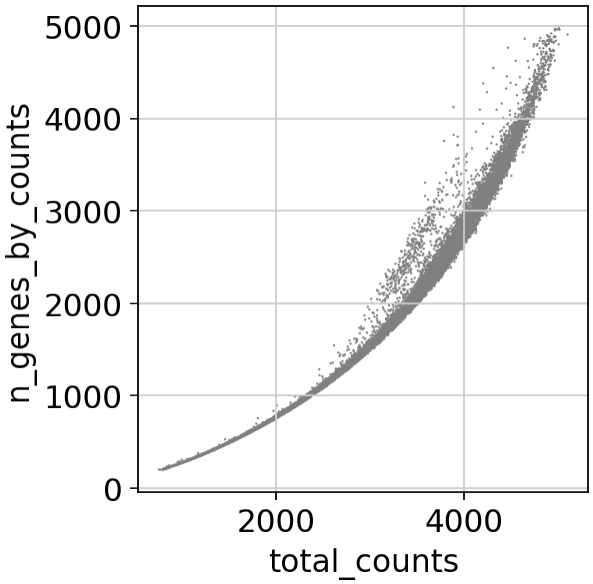
```
filtered out 1 cells that have less than 200 genes expressed
filtered out 1675 genes that are detected in less than 3 cells
```

In [47]:
```
adata.var['mt'] = adata.var_names.str.startswith('MT-')  # annotate the group of mitochondrial genes as 'mt'
sc.pp.calculate_qc_metrics(adata, qc_vars=['mt'], percent_top=None, log1p=False, inplace=True)
```

In [48]:
```
#plot the result of the previous calculation
sc.pl.violin(adata, ['n_genes_by_counts', 'total_counts', 'pct_counts_mt'],
             jitter=0.4, multi_panel=True)
```



In [49]:
```
#plot genes by counts against total counts per cell
sc.pl.scatter(adata, x='total_counts', y='n_genes_by_counts')
```



In [50]:
```
#total-count normalize (library-size correct) the data matrix  to 10,000 reads per cell,
#so that counts become comparable among cells.
sc.pp.normalize_total(adata, target_sum=1e4)
```

normalizing counts per cell
finished (0:00:00)

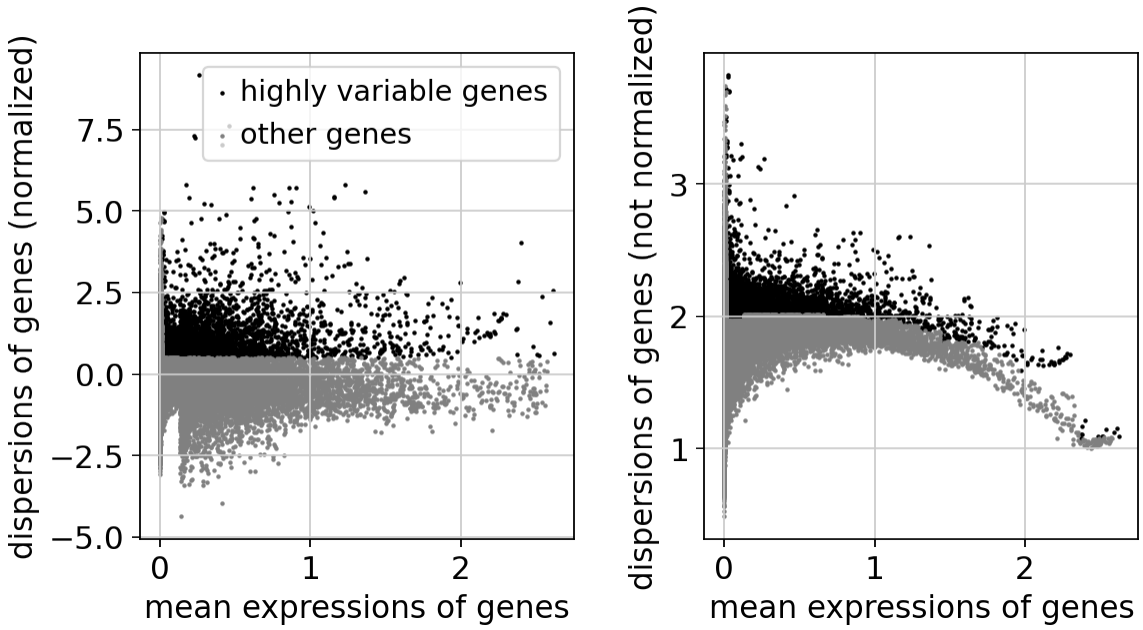
In [51]:
```
#log transform the data
sc.pp.log1p(adata)
```

In [52]:
```
#identify highly variable genes
```

```
sc.pp.highly_variable_genes(adata, min_mean=0.0125, max_mean=3, min_disp=0
.5)
```

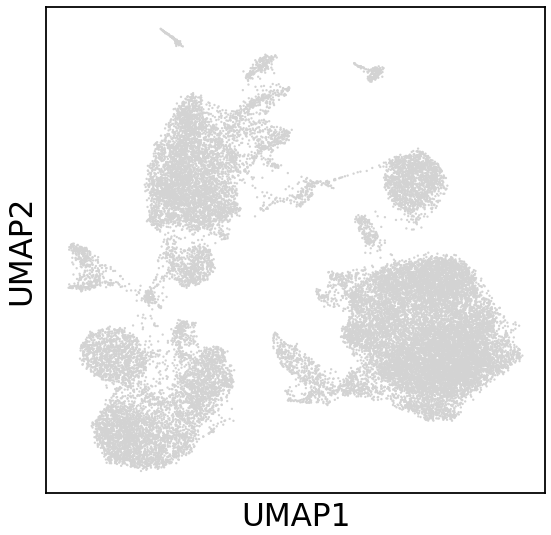
```
extracting highly variable genes
    finished (0:00:06)
--> added
    'highly_variable', boolean vector (adata.var)
    'means', float vector (adata.var)
    'dispersions', float vector (adata.var)
    'dispersions_norm', float vector (adata.var)
```

In [53]:
```
#plot the highly variable genes
sc.pl.highly_variable_genes(adata)
```

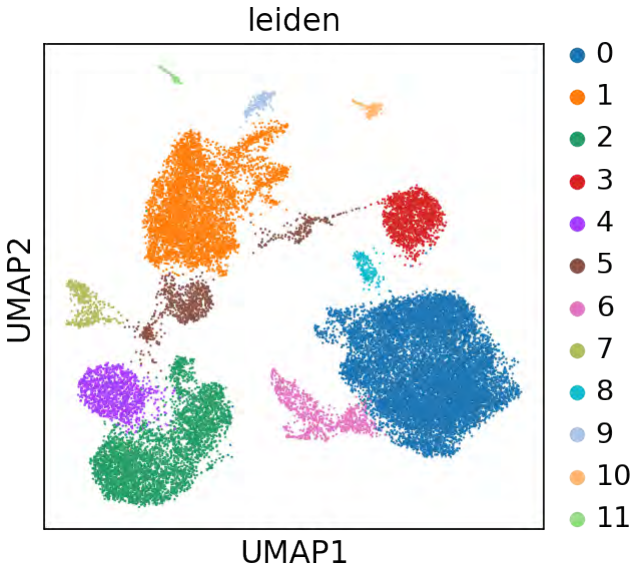


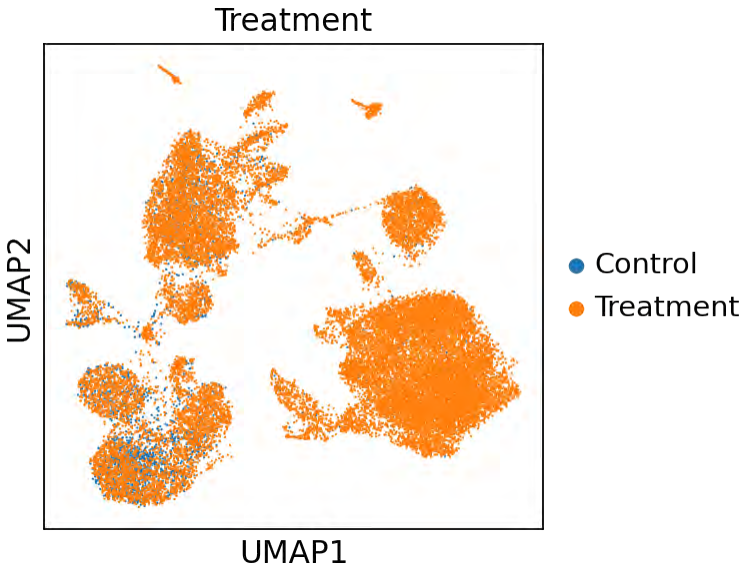
UMAPs and marker genes

In [54]:
```
#plot a basic UMAP
sc.pl.umap(adata)
```

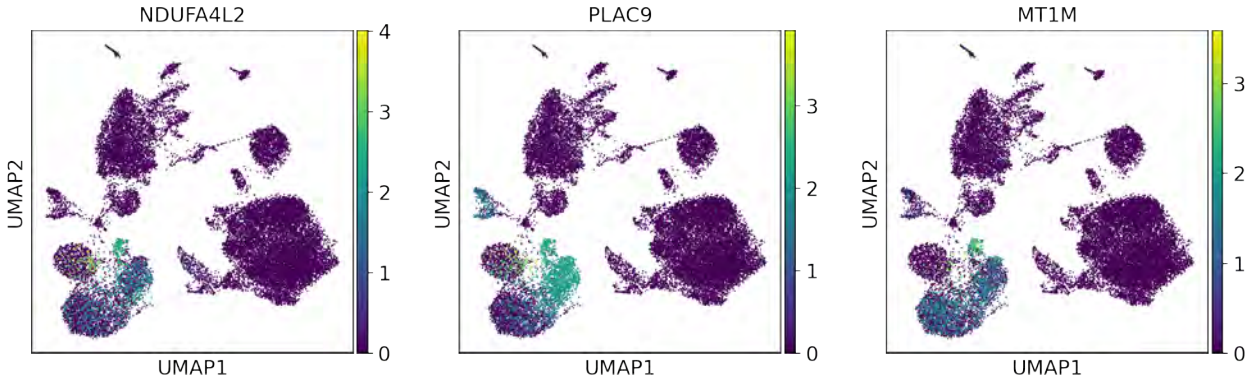


In [55]:
```
#and plot the UMAP using louvain1.0 coloring scheme:
sc.pl.umap(adata, color=['leiden'])
sc.pl.umap(adata, color=['Treatment'])
```

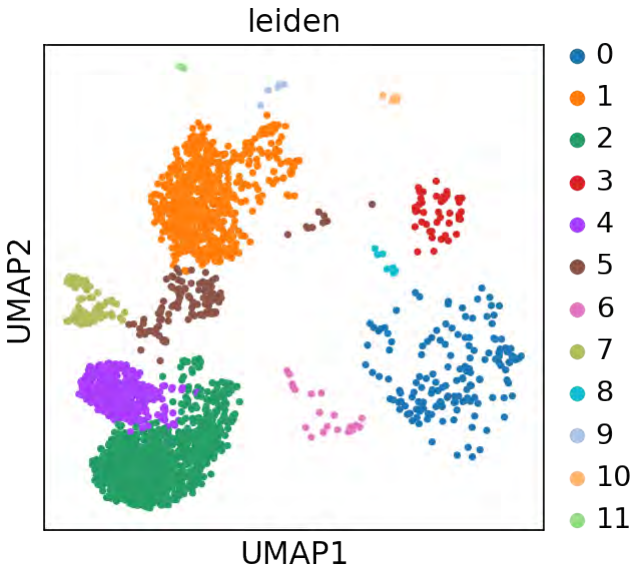


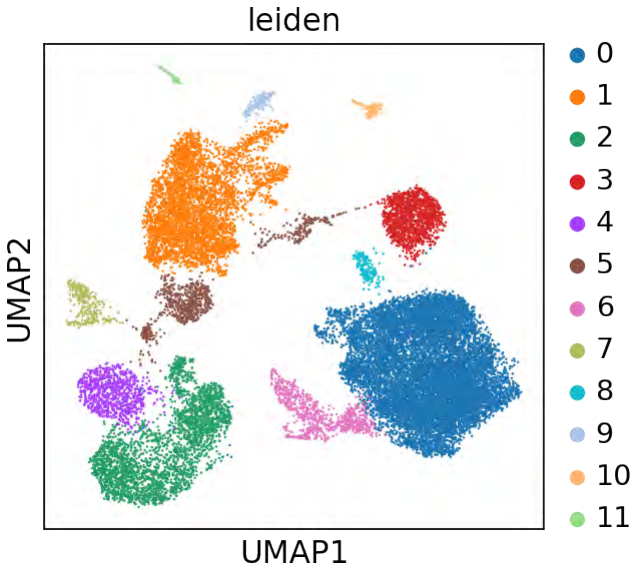


In [56]:
```
#plot the UMAP coloring for a few genes of interest
#note: replace the text below with your specific genes of interest
sc.pl.umap(adata, color=['NDUFA4L2','PLAC9','MT1M'])
```

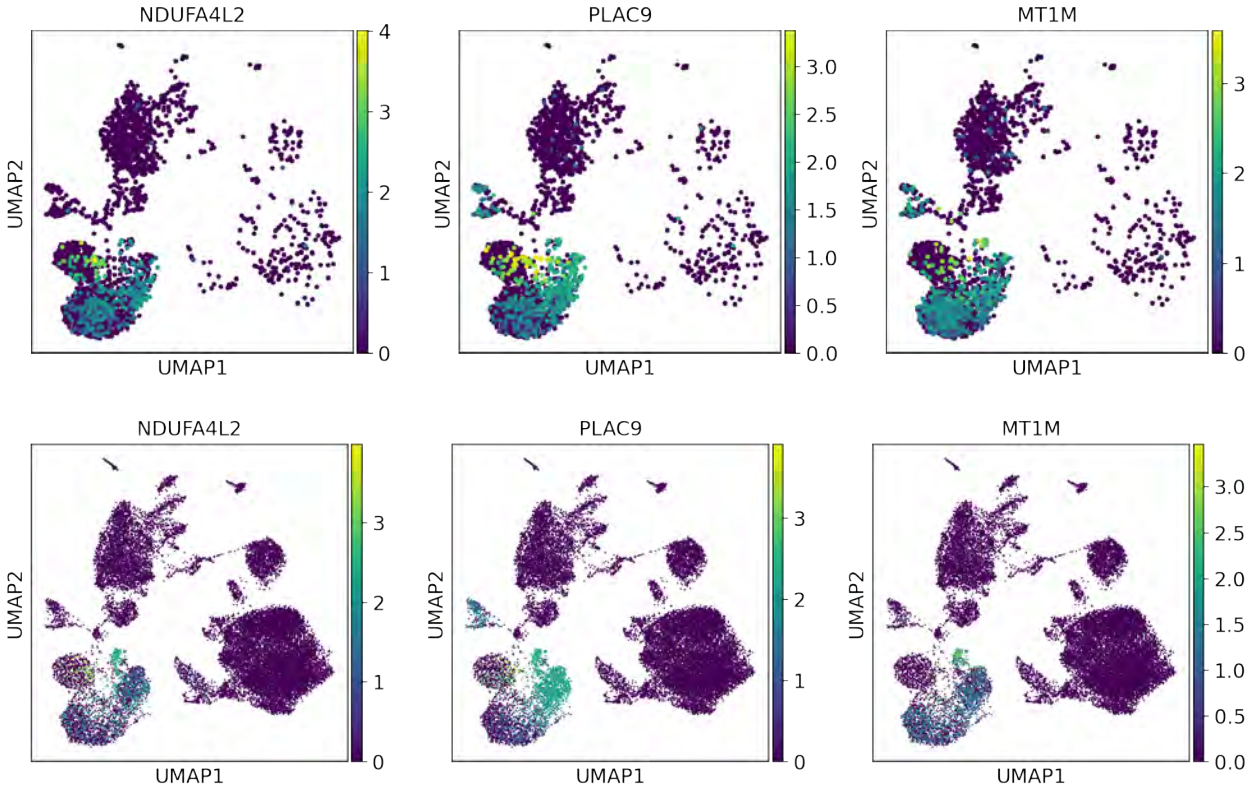


In [57]:
```
#plotting leiden clustering for the Control and the Treatment (Aortic aneurysm)
sc.pl.umap(adata[adata.obs['Treatment'].isin(['Control'])], color='leiden'
)
sc.pl.umap(adata[adata.obs['Treatment'].isin(['Treatment'])], color='leide
n')
```



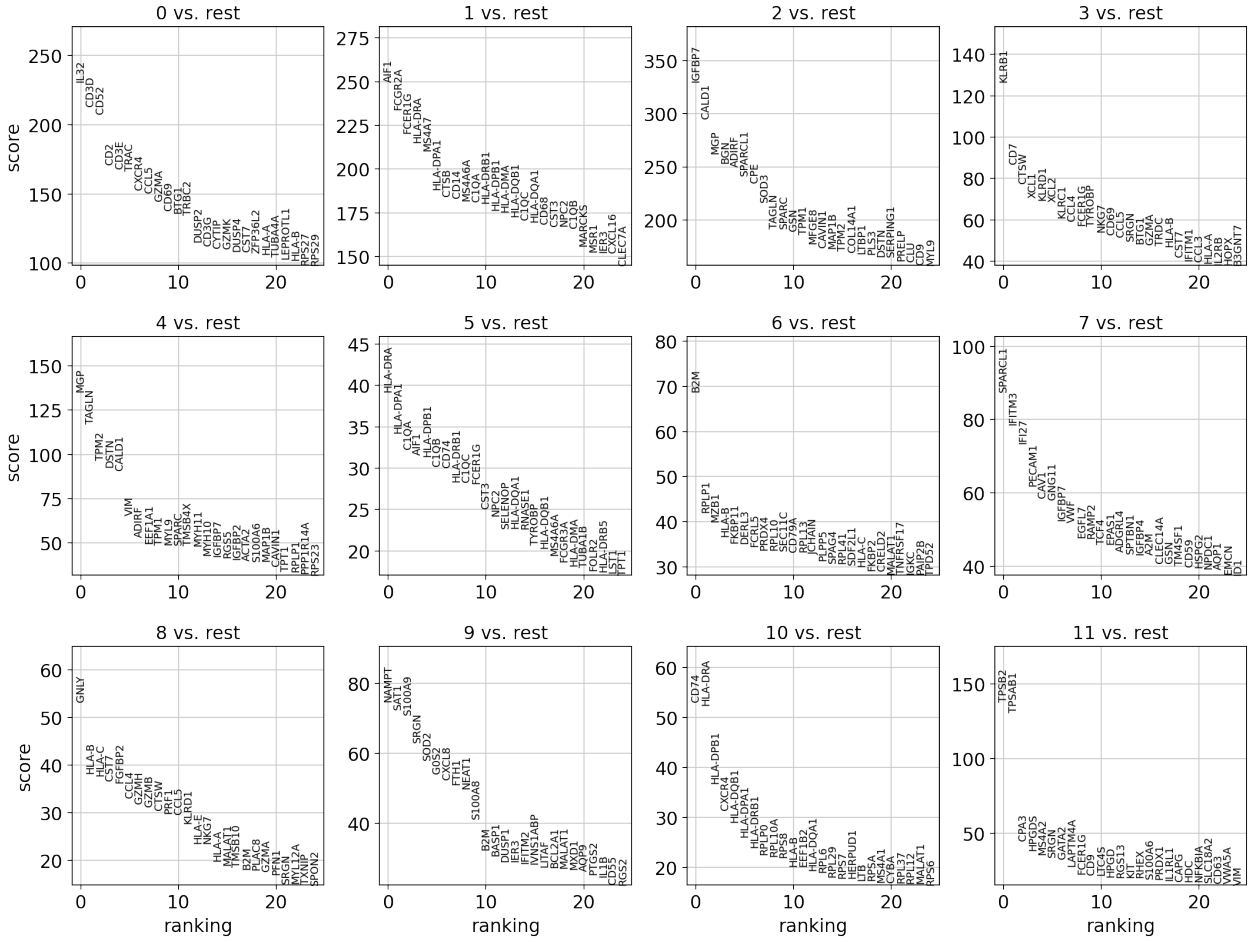


In [58]:
```
#plotting genes of interest for the Control and the Treatment (Aortic aneurysm) for genes of interest
sc.pl.umap(adata[adata.obs['Treatment'].isin(['Control'])], color=['NDUFA4
L2','PLAC9','MT1M'])
sc.pl.umap(adata[adata.obs['Treatment'].isin(['Treatment'])], color=['NDUF
A4L2','PLAC9','MT1M'])
```



In [59]:
```
#identifying marker genes using leiden clusters and a t-test
sc.tl.rank_genes_groups(adata, 'leiden', method='t-test')
sc.pl.rank_genes_groups(adata, n_genes=25, sharey=False)
```

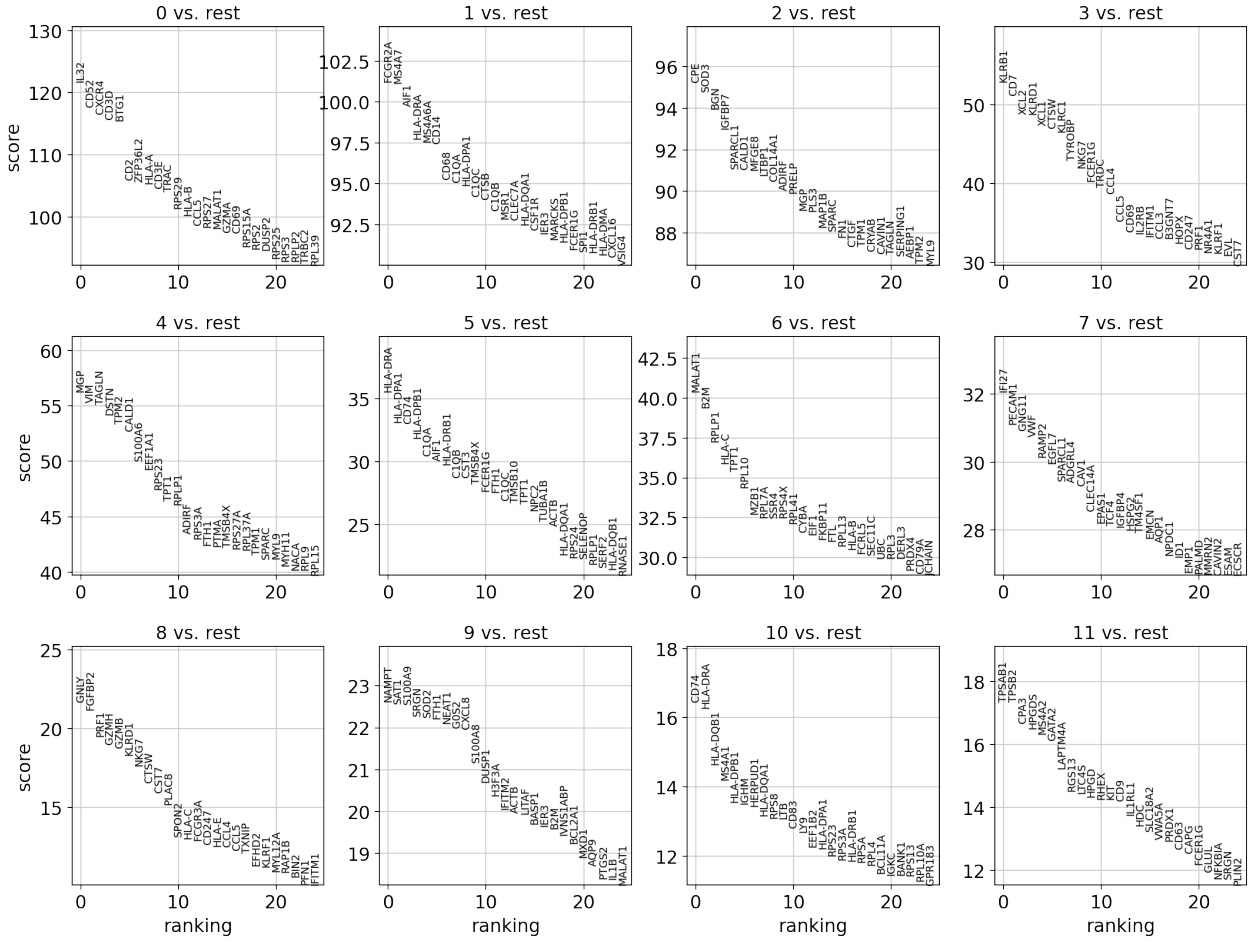
ranking genes
finished: added to `.uns['rank_genes_groups']`
'names', sorted np.recarray to be indexed by group ids
'scores', sorted np.recarray to be indexed by group ids
'logfoldchanges', sorted np.recarray to be indexed by group ids
'pvals', sorted np.recarray to be indexed by group ids
'pvals_adj', sorted np.recarray to be indexed by group ids (0:00:42)



In [60]:
```
#write this out to a file
adata.write('leiden_ttest_markers.txt')
```

In [61]:
```
#identifying marker genes using wilcoxon rank-sum testt
sc.tl.rank_genes_groups(adata, 'leiden', method='wilcoxon')
sc.pl.rank_genes_groups(adata, n_genes=25, sharey=False)
```

ranking genes
finished: added to `.uns['rank_genes_groups']`
'names', sorted np.recarray to be indexed by group ids
'scores', sorted np.recarray to be indexed by group ids
'logfoldchanges', sorted np.recarray to be indexed by group ids
'pvals', sorted np.recarray to be indexed by group ids
'pvals_adj', sorted np.recarray to be indexed by group ids (0:01:27)

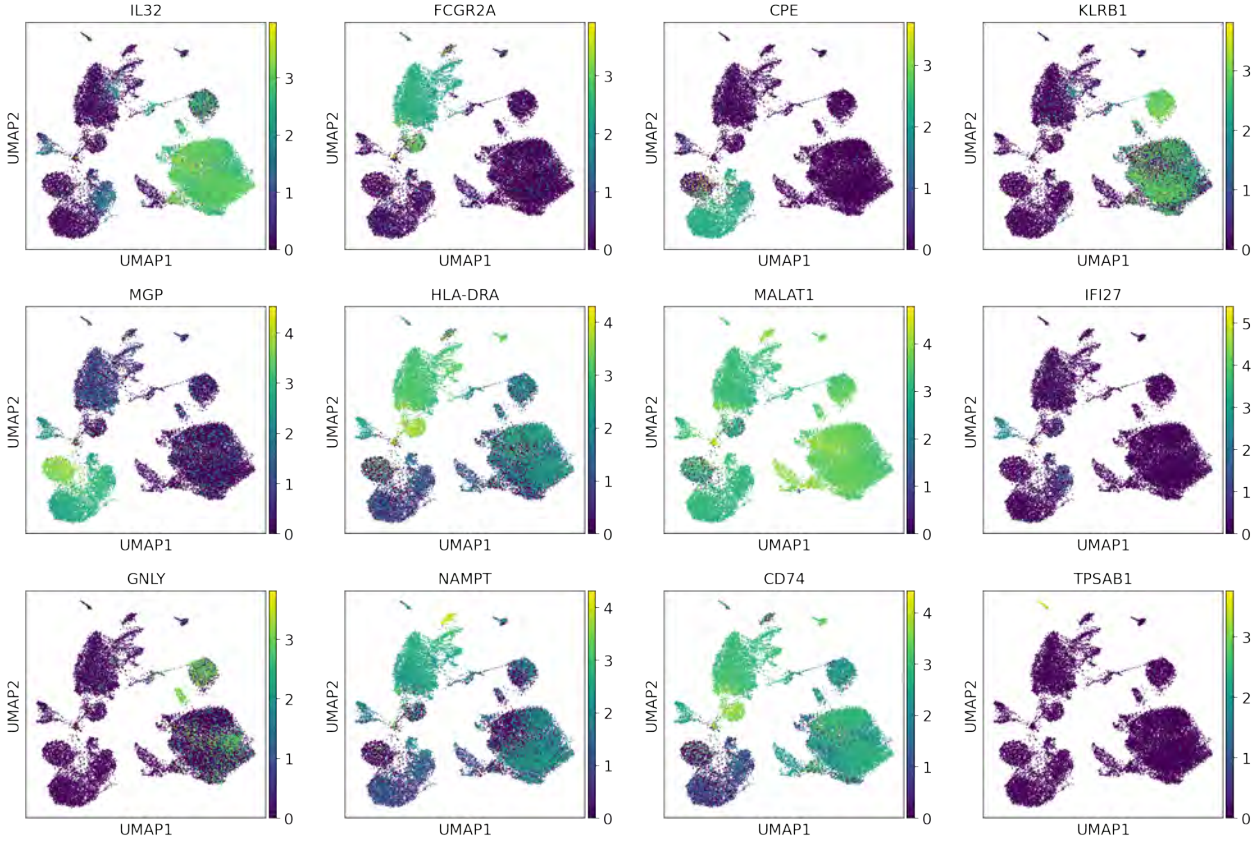


In [62]:
```
#write this out to a file
adata.write('leiden_wilcoxon_markers.txt')
```

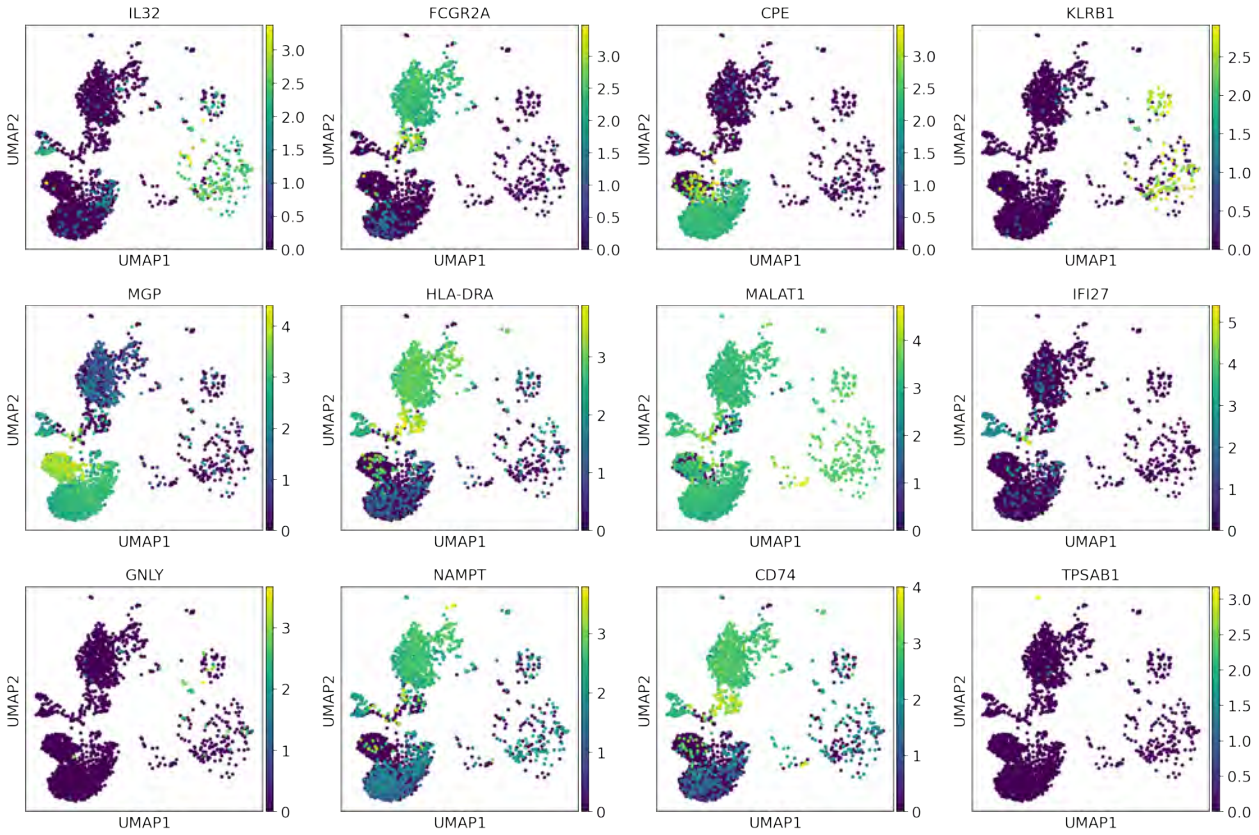
In [63]:
```
#defining the first ranked marker gene from wilcoxon rank sum testing
marker_genes = ['IL32','FCGR2A','CPE','KLRB1','MGP','HLA-DRA','MALAT1','IFI27','GNLY','NAMPT','CD74'
                ,'TPSAB1']
```

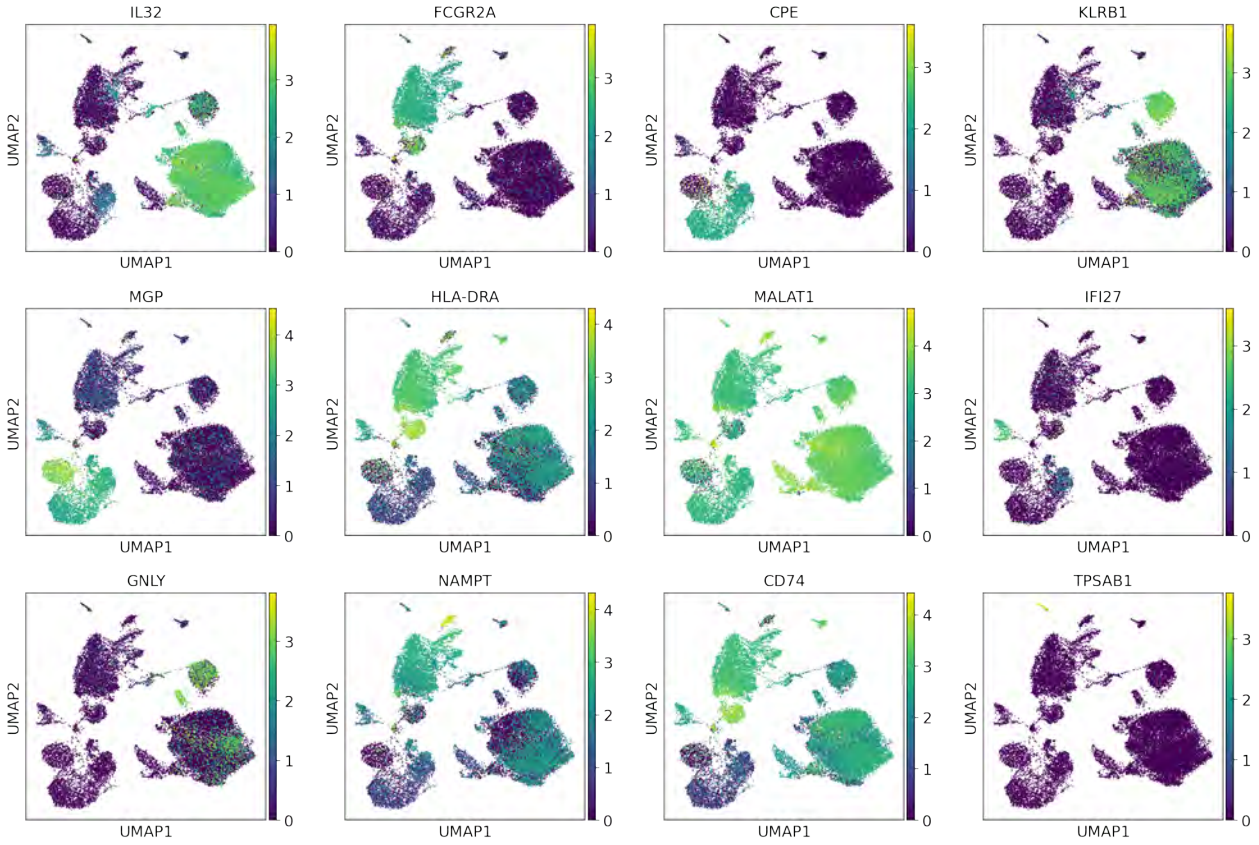
Examining the marker genes in the dataset

In [64]:
```
#plotting a umap of these marker genes
sc.pl.umap(adata, color=marker_genes)
```



In [65]:
```
#and doing this with respect to treatment/control cells
sc.pl.umap(adata[adata.obs['Treatment'].isin(['Control'])], color=marker_genes)
sc.pl.umap(adata[adata.obs['Treatment'].isin(['Treatment'])], color=marker_genes)
```





In [66]:
```
#show the top 5 markers for each cluster!
pd.DataFrame(adata.uns['rank_genes_groups']['names']).head(5)
```

Out[66]:

	0	1	2	3	4	5	6	7	8	9	10	11
0	IL32	FCGR2A	CPE	KLRB1	MGP	HLA-DRA	MALAT1	IFI27	GNLY	NAMPT	CD74	TPSAB1
1	CD52	MS4A7	SOD3	CD7	VIM	HLA-DPA1	B2M	PECAM1	FGFBP2	SAT1	HLA-DRA	TPSB2
2	CXCR4	AIF1	BGN	XCL2	TAGLN	CD74	RPLP1	GNG11	PRF1	S100A9	HLA-DQB1	CPA3
3	CD3D	HLA-DRA	IGFBP7	KLRD1	DSTN	HLA-DPB1	HLA-C	VWF	GZMH	SRGN	MS4A1	HPGDS
4	BTG1	MS4A6A	SPARCL1	XCL1	TPM2	C1QA	TPT1	RAMP2	GZMB	SOD2	HLA-DPB1	MS4A2

In [67]:
```
#make a more detailed table, including p-values
result = adata.uns['rank_genes_groups']
groups = result['names'].dtype.names
pd.DataFrame(
    {group + '_' + key[:1]: result[key][group]
    for group in groups for key in ['names', 'pvals']}).head(5)
```

Out[67]:

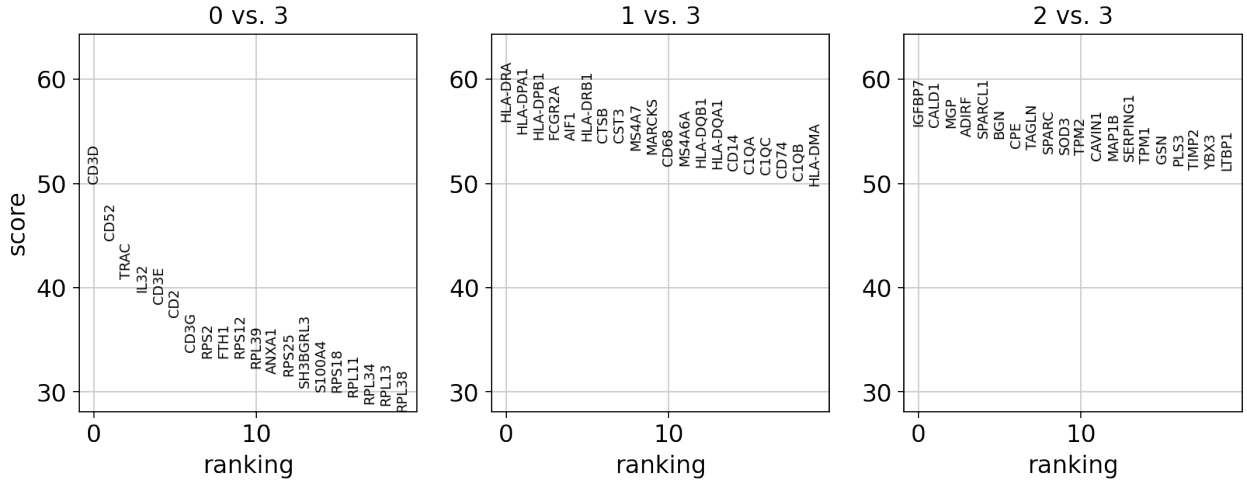
	0_n	0_p	1_n	1_p	2_n	2_p	3_n	3_p	4_n	4_p	...	7_n	7_p	8_
0	IL32	0.0	FCGR2A	0.0	CPE	0.0	KLRB1	0.0	MGP	0.0	...	IFI27	9.019634e-226	GN

1	CD52	0.0	MS4A7	0.0	SOD3	0.0	CD7	0.0	VIM	0.0	...	PECAM1	1.612090e-212	FGFBI
2	CXCR4	0.0	AIF1	0.0	BGN	0.0	XCL2	0.0	TAGLN	0.0	...	GNG11	3.551387e-210	PR
3	CD3D	0.0	HLA-DRA	0.0	IGFBP7	0.0	KLRD1	0.0	DSTN	0.0	...	VWF	8.100193e-208	GZN
4	BTG1	0.0	MS4A6A	0.0	SPARCL1	0.0	XCL1	0.0	TPM2	0.0	...	RAMP2	8.744787e-200	GZN

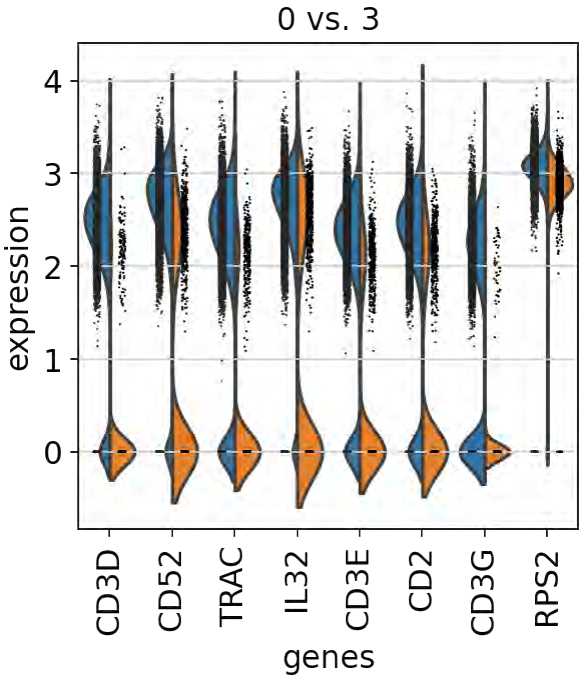
5 rows × 24 columns

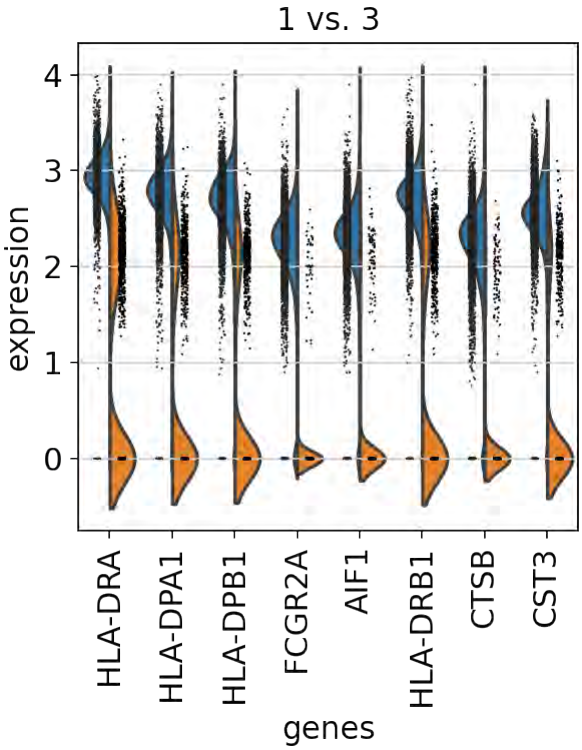
In [68]:
```
#comparing two clusters (0 and 1)
sc.tl.rank_genes_groups(adata, 'leiden', groups=['0','1','2'], reference='3', method='wilcoxon')
sc.pl.rank_genes_groups(adata, groups=['0','1','2'], n_genes=20)
```

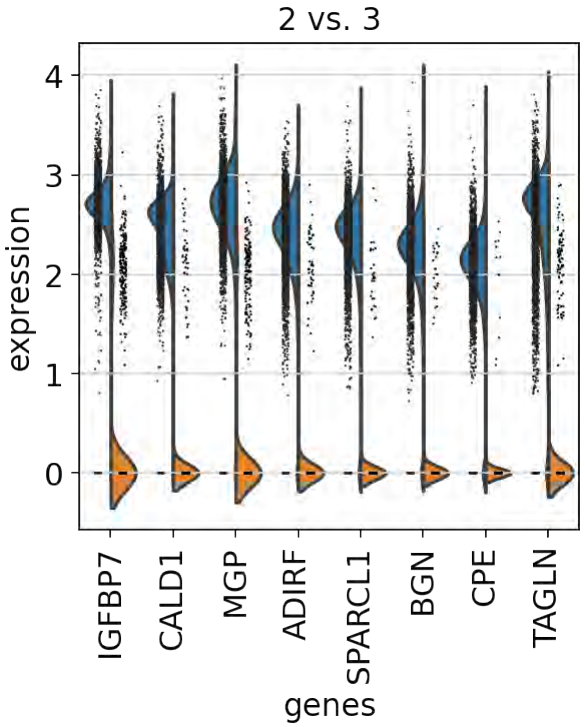
ranking genes
finished: added to `.uns['rank_genes_groups']`
'names', sorted np.recarray to be indexed by group ids
'scores', sorted np.recarray to be indexed by group ids
'logfoldchanges', sorted np.recarray to be indexed by group ids
'pvals', sorted np.recarray to be indexed by group ids
'pvals_adj', sorted np.recarray to be indexed by group ids (0:00:44)



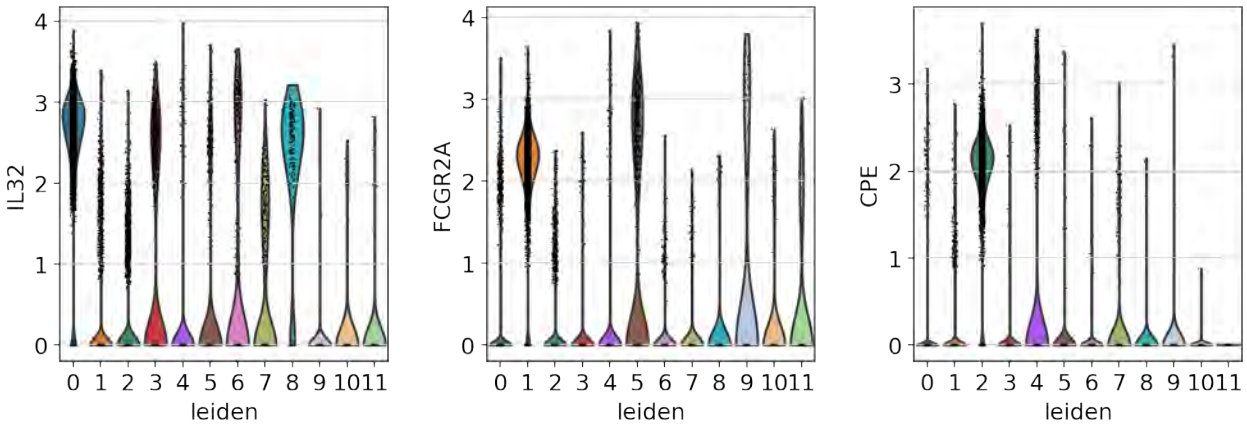
In [69]:
```
#If we want a more detailed view for a certain group, use sc.pl.rank_genes_groups_violin.
sc.pl.rank_genes_groups_violin(adata, groups=['0','1','2'], n_genes=8)

#note that you can replace the "groups" and "n_genes" variables as you please here....
```


0 vs. 3
expression
0
1
2
3
4
CD3D
CD52
TRAC
IL32
CD3E
CD2
CD3G
RPS2
genes


1 vs. 3
expression
0
1
2
3
4
HLA-DRA
HLA-DPA1
HLA-DPB1
FCGR2A
AIF1
HLA-DRB1
CTSB
CST3
genes



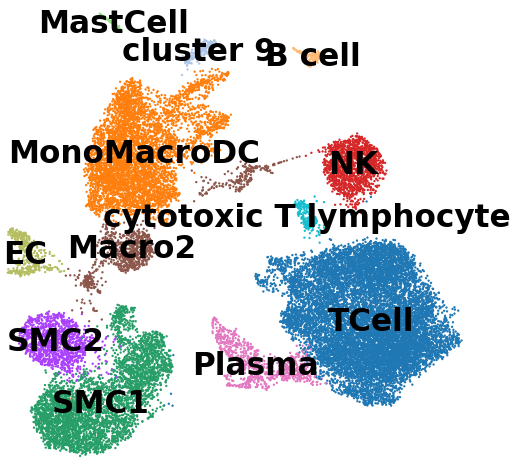
In [70]:
```
#comparing different genes across the groups
sc.pl.violin(adata, ['IL32','FCGR2A','CPE'], groupby='leiden')
```



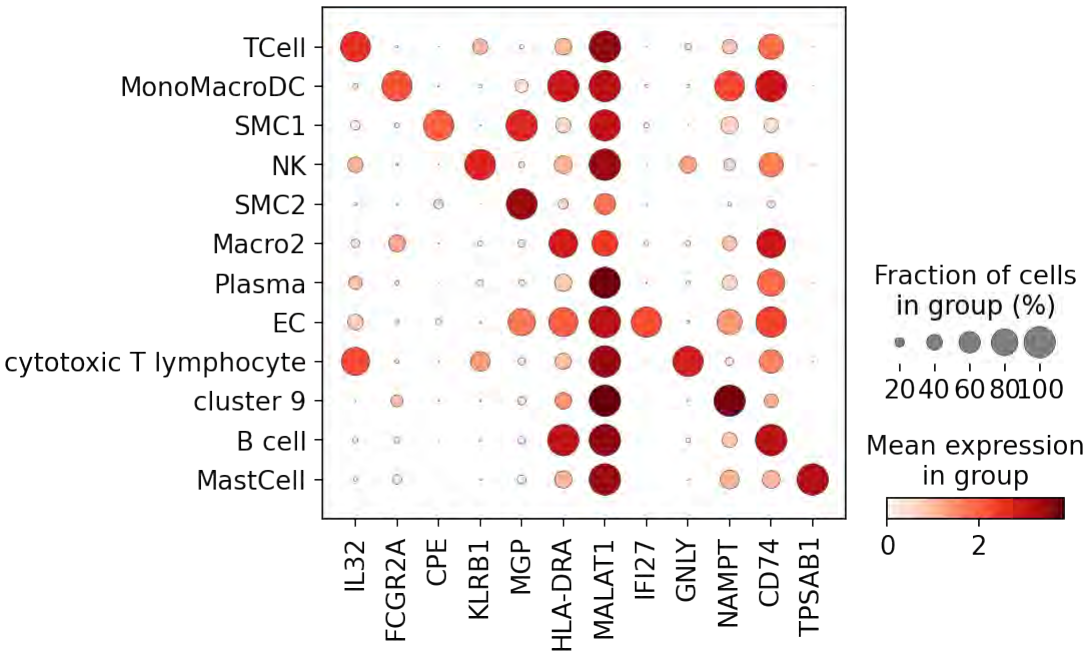
In [72]:
```
#naming the clusters by their putative identities
#see https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.120.046528

new_cluster_names = [
    'TCell',
    'MonoMacroDC',
    'SMC1',
    'NK',
    'SMC2',
    'Macro2',
    'Plasma',
    'EC',
    'cytotoxic T lymphocyte',
    'cluster 9',
    'B cell',
    'MastCell']
adata.rename categories('leiden', new cluster names)
```

```
Omitting rank_genes_groups/names as old categories do not match.
Omitting rank_genes_groups/scores as old categories do not match.
Omitting rank_genes_groups/pvals as old categories do not match.
Omitting rank_genes_groups/pvals_adj as old categories do not match.
Omitting rank_genes_groups/logfoldchanges as old categories do not match.
```

In [73]:
```
#making a umap with cluster names
sc.pl.umap(adata, color='leiden', legend_loc='on data', title='', frameon=
False)
```

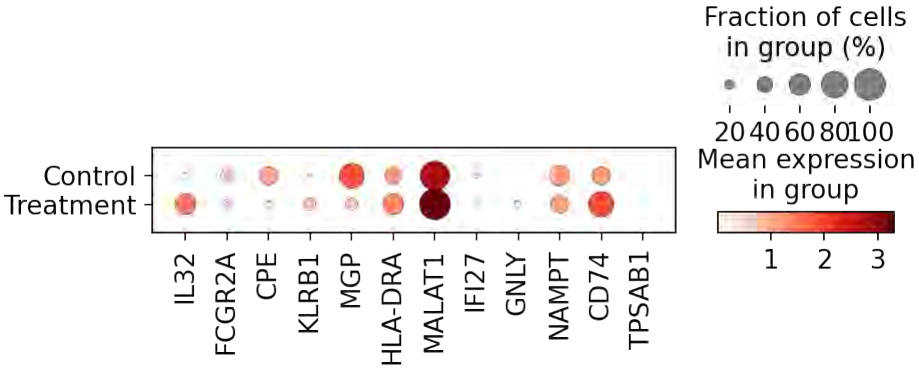


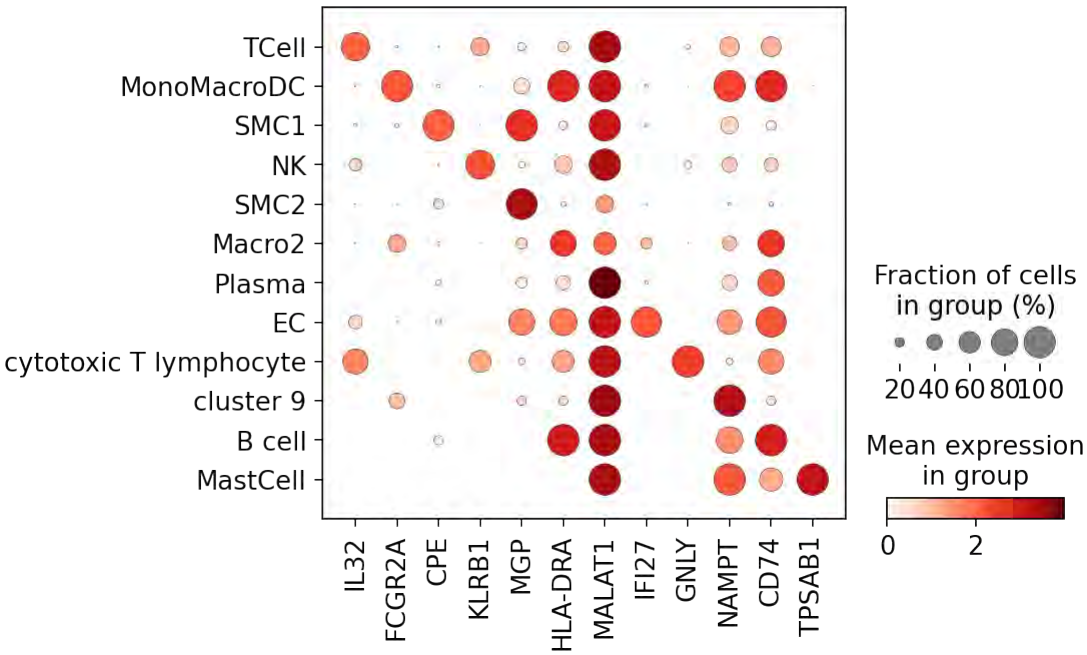
In [76]:
```
#making a dotplot of the top marker genes
sc.pl.dotplot(adata, marker_genes, groupby='leiden');
```

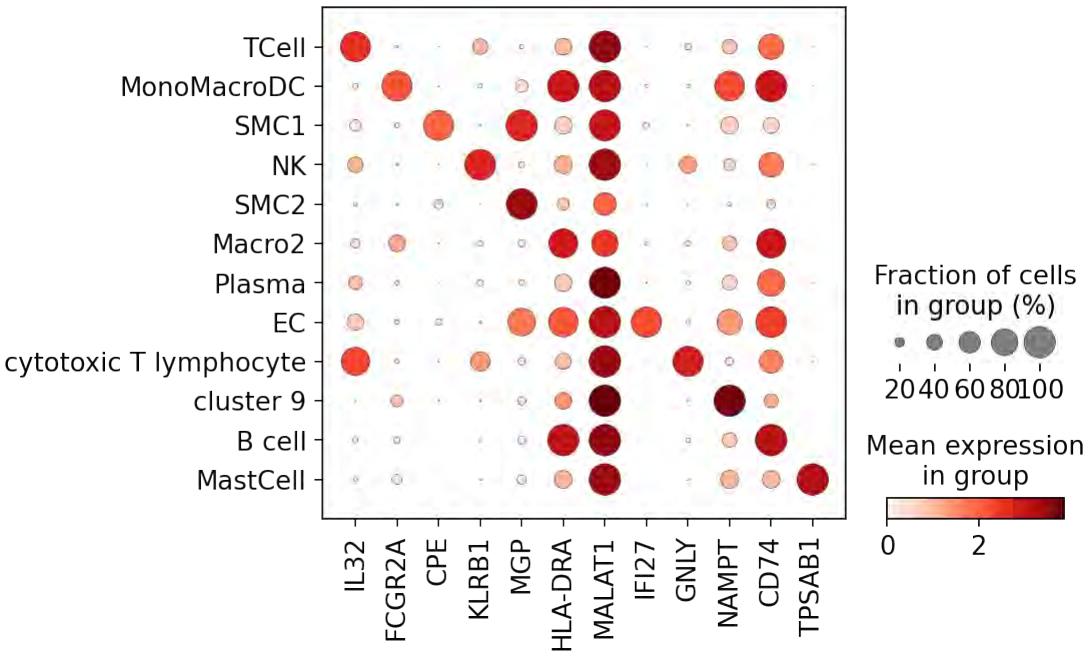


In [78]:
```
#and comparing this dotplot by Treatment vs Control
sc.pl.dotplot(adata, marker_genes, groupby='Treatment');
```

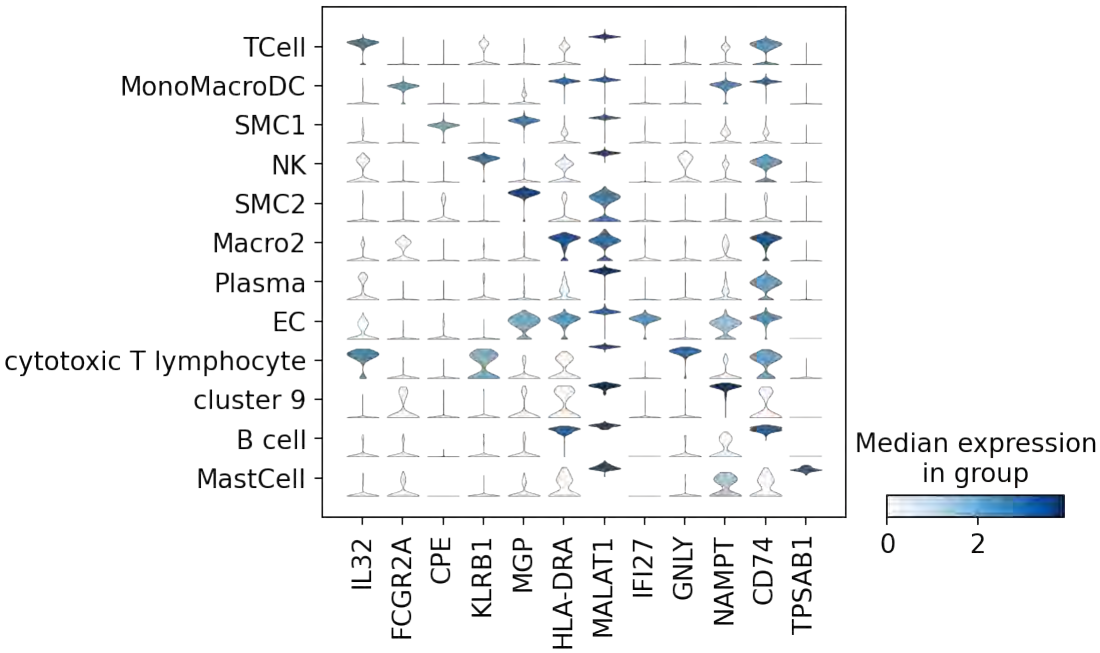
```
sc.pl.dotplot(adata[adata.obs['Treatment'].isin(['Control'])], marker_gene
s, groupby='leiden')
sc.pl.dotplot(adata[adata.obs['Treatment'].isin(['Treatment'])], marker_ge
nes, groupby='leiden')
```







In [79]:
```
#and a very compact violin plot
sc.pl.stacked_violin(adata, marker_genes, groupby='leiden', rotation=90);
```



In [80]:
```
#examine the adata object one more time
adata
```

Out[80]:
```
AnnData object with n_obs × n_vars = 27224 × 21643
    obs: 'Sample', 'Treatment', 'batch', 'n_genes', 'n_genes_by_counts', 'total_counts', 'total_counts_mt', 'pct_count
```

```
s_mt', 'leiden'
    var: 'mt', 'gene_ids', 'n_cells_by_counts', 'mean_counts', 'pct_dropout_by_counts', 'total_counts', 'highly_variable', 'means', 'dispersions', 'dispersions_norm', 'n_cells'
    uns: 'Sample_colors', 'Treatment_colors', 'hvg', 'leiden', 'neighbors', 'pca', 'umap', 'log1p', 'leiden_colors', 'rank_genes_groups'
    obsm: 'X_pca', 'X_umap'
```

Listing our output files

In [5]:
```
#let's list our files that we've made:
!ls -lah
```

```
total 9.2G
drwxrwxr-x 4 welder-user users 4.0K Sep 22 21:02 .
drwxrwxr-x 4 welder-user users 4.0K Sep 21 15:10 ..
drwxrwxr-x 2 jupyter     users 4.0K Sep 21 15:12 data
-rw-rw-r-- 1 welder-user users   72 Sep 22 20:46 .delocalize.json
-rw-rw-r-- 1 welder-user users  15M Sep 22 21:02 human_aorta_cumulus_scanpy.ipynb
-rw-rw-r-- 1 jupyter     users  136 Sep 21 16:35 human_aorta_cumulus_scanpy-requirements.txt
drwxrwxr-x 2 jupyter     users 4.0K Sep 21 15:10 .ipynb_checkpoints
-rw-rw-r-- 1 jupyter     users 4.6G Sep 21 17:01 leiden_ttest_markers.txt
-rw-rw-r-- 1 jupyter     users 4.6G Sep 21 17:05 leiden_wilcoxon_markers.txt
```

In [ ]:
```

```